

(19)



Europäische Patentamt

European Patent Office

Offic Europe n d s br v ts



(11)

EP 0 721 286 A2

(12)

## EUROPEAN PATENT APPLICATION

(43) Date of publication:

10.07.1996 Bulletin 1996/28

(51) Int Cl. 6: H04N 7/30

(21) Application number: 96100033.8

(22) Date of filing: 03.01.1996

(84) Designated Contracting States:

DE FR GB IT NL

(30) Priority: 09.01.1995 JP 1057/95

25.01.1995 JP 9712/95

(71) Applicant: MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.

Kadoma-shi, Osaka-fu, 571 (JP)

(72) Inventors:

- Chong, Siew T.  
Singapore 1954 (SG)
- Shen, Sheng M.  
Singapore 2264 (SG)
- Kohr, Chee S.  
Singapore 2056 (SG)

(74) Representative: Eisenführ, Speiser & Partner

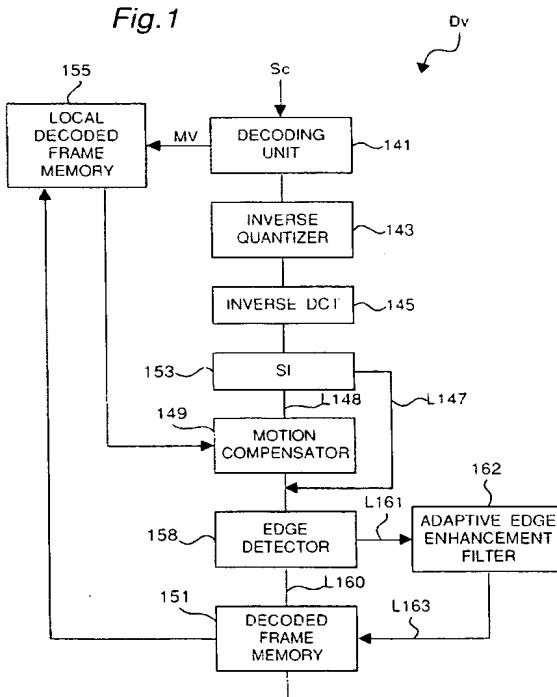
Martinistraße 24

28195 Bremen (DE)

### (54) Video signal decoding apparatus with artifact reduction

(57) A video signal decoding apparatus (Dv) according to the present invention includes a decoding unit (141) for applying a variable length and run-length decoding to the compressed video signal (Sc) to reconstruct blocks of quantized coefficients. An inverse quantizer (143) is provided for inverse quantizing blocks of reconstructed quantized coefficients. An inverse DCT (145) transforms the inverse quantized blocks of reconstructed quantized coefficients to produce a decoded block. An edge block detector (153, 149 and 158) determines whether said decoded block is classified as an edge block or not. An adaptive edge enhancement filter (162) removes the distortion in the decoded block when the edge block detector (153) determines as an edge block. Additionally, a ringing noise filter is disclosed for the same decoding apparatus.

Fig. 1



EP 0 721 286 A2

## D scription

### BACKGROUND OF THE INVENTION

#### 1. Fi ld of th Inv ntion

The present invention relates to a video signal that have been coded using transform coding technique and, more particularly, to a video signal decoding apparatus which can remove the ringing noise component included in that coded video signal for improved picture quality.

The present invention further relates to a video signal decoding apparatus for removing ringing noise components around the high frequency areas of the decoded digital data and, more particularly, to a ringing noise removing apparatus especially suitable for removing ringing digital video data that are reconstructed from decompressed video data. Therefore, there is little or no degradation in quality in other areas of the reconstructed digital video that had been processed by the apparatus mentioned above, resulting in a visually more pleasant and noise reduced output.

#### 2. Description of the Prior Art

Digital transmission and storage of video signal and its associated audio has attracted a lot of attention recently because of the superior quality and greater flexibility compared to the analog signal. However, as digital video picture requires a large amount of data for its representation, this creates a constraint on its widespread use. Coding (compression) technique were developed to reduce the amount of data needed to represent the video signal.

Various techniques of compression have been explored for the compression of video signal. These methods are usually lossy in nature, that is, they attempt to use less amount of bits to represent the most important information present in the data, such that normal viewers of the decoded video signal will find it difficult to detect the difference between the decoded video signal and the original non-coded signal.

One of the popular form of compression is to apply transform coding to the digital signal to transform the digital signal from the spatial domain to another domain (for example the frequency domain). The idea behind transform coding is to make use of the energy compaction property of the transformation to decorrelate the image pixel so that redundancy in the image can be removed more efficiently in the transformed domain.

Various transformation techniques has been explored for the removal of redundancy in the spatial domain, and Discrete Cosine Transform (DCT) is found to be the best transformation from efficient and implementation complexity point of view. Thus, DCT has been the basic core technique of a few recently established International Standardization Organization (ISO) standards for the compression of images. which include JPEG

(Joint Photographic Expert Group), MPEG (Moving Picture Expert Group) Phase-1, MPEG Phase-2, h.261 (CCITT standards, used mainly for teleconferencing).

To compress an image using the DCT method, the

5 image are divided into square blocks of pixel (says 8 x 8 pixel block) and each of these blocks is then transformed using DCT into the transformed block (8 x 8 also) which contains the DCT coefficients. As the neighboring pixel in an images are in general highly correlated, this  
 10 enables DCT to transform image pixel into DCT coefficients that are energy concentrated in only a few of the coefficients around the top-left corner of the transformed block, which is the low frequency region. These transformed blocks are then quantified into quantized coefficient  
 15 and then run-length and variable length coded to further reduce the statistical redundancy present into the run-length coded data.

The energy compaction property of DCT can be easily applied in lock of pixel that are highly correlated.

20 However, in the case of edge block, i.e. block which contains parts of edges that have sharp change in intensity across the edge, the compaction of signal energy cannot be achieved efficiently.

With reference to Figs. 7A, 7B, 8A, and 8B, the ef-

25 fect of applying DCT to blocks of pixel of different natures is described later. In Figs. 7A and 7B, one example of a non edge block (8 x 8 pixel) and its corresponding transformed block are shown, respectively. As best shown in Fig. 7B, the energy is concentrated on the top-left corner of the 8 x 8 pixel block as a result of transformation. Quantization, run-length coding and variable length coding of this transform block will produce little bits, and hence an efficient compression can be achieved.

30 In Figs. 8A and 8B, one example of an edge block (8 x 8 pixel) and its corresponding transformed block are shown, respectively. As best shown in Fig. 8B, the transformed edge block (Fig. 8A) has DCT coefficients that are much more randomly distributed, when compared  
 35 with the edged block. Such transform block is more difficult to quantized, run-length coded and variable length coded and hence the compression of the edge block cannot be done efficiently.

Furthermore, in order to compress more bits, the

40 Quantization parameters are designed to preserve the low frequency coefficients at the upper left corner of the block of 8 x 8 pixel, this design will cause more error to be generated for edge block when the edge block is decoded by the decoder. The error will cause a ringing effect (corona effect) appearing around the edge of the reconstructed block. This effect causes noise component in the high frequency area of the decoded video signal, and such noise is referred to as a "ringing noise".

45 With reference to Figs. 9A, 9B, 10A, and 10B, the

50 ringing effect are described specifically. In Figs. 9A and 9B, one example of an ideal vertical edge block and its corresponding waveform are shown, respectively. This ideal vertical edge block is Discrete Cosine Transformed

(DCTed), quantized, and then inverse quantized, inverse DCTed to produce the decoded block.

In Figs. 10A and 10B, thus decoded block and its corresponding wave form are shown, respectively. It can be seen that ripples R appear on both side, enclosed by circles, of the edge constituting the ringing effects. As edge information is very sensitive to the human eyes compared to other high frequency pixel region, distortion around the edges, i.e. the ringing noise, represents a source of irritation to the viewers.

To solve this problem, an efficient method of removing the ringing effect around the edge is disclosed in the Japanese Laid-open Patent Publication No. H6-292173 published October 18, 1994, by Siew Tan Chong and entitled and assigned to the same assignee of the present invention. In this previously disclosed method, the decoded block is subject to an edge detection process which classified the said decoded block into edge block and non-edge block, and then subjecting the edge block to adaptive smoothing process to reduce the ringing effect.

In general, DCT has been found to be effective for the compression of digital images. However, when the block of pixel contains edge which has sharp change in intensity, the energy compaction property of DCT is no longer valid. This will result in the decoded edge block with ringing effect around the edges in the block. Such distortion is annoying to the viewers as edge provides important information to the viewers.

Such ringing effect can be removed by performing smoothing operation to the decoded block using smoothing filter. However, well known soothing filters, like the average filter and median filter, also remove the contrast of the edges, thus creating a blurring effect to the originally sharp edge. Hence, these smoothing filters are not suitable for the removal of ringing effect.

Furthermore, in order to adaptively filter only those block which contain edge information, an edge detection algorithm needs to be designed to differentiate between an edge block from a non edge block. Most of the currently available edge detection algorithm involves many mathematical computation, for example, the use of squaring function, square root function or the use of surrounding blocks to help make decision. This increases the hardware complexity necessary to implement an edge enhancement filter.

In addition to the above, over the past few years, many video compression methods had been proposed and standardized for the delivery of digital video such as MPEG-1, MPEG-2, JPEG, H.261 ect., and other proprietary standards. Most of the compression standards are lossy in nature in order to achieve the high compression rates. Due to this lossy nature, artifacts appear when the compressed digital video data is decompressed to reconstruct the original digital video data.

In Figs. 18 and 19, examples of frequency response of high frequency areas of digital video data before compression and after reconstruction are shown, respec-

tively. As best shown in Fig. 19, ringing noises are observed near the high frequency area of the data reconstructed from the compressed state.

To reduce or remove these artifacts, or ringing noise, postprocessing of the decompressed digital video is required. However, most of the current post-processing methods are IIR (Infinite Impulse Response) or FIR (Finite Impulse Response) based filters which are either too complicated and costly to implement or result in degradation of the reconstructed video quality which occur when video data not affected by noise are altered significantly during the removal of the artifacts.

As described above, there is problems with current compression methods which causes artifacts in the reconstructed digital video data from the decompressed data. One of the more serious artifacts is the appearance of ringing noise at and around the high frequency areas of a reconstructed digital video data, as shown in Fig. 19. This usually occurs when one of the process in the compression methods includes DCT (Discrete Cosine Transform), a lossy compression process.

An apparatus had been invented which will remove the ringing noise around the high frequency areas from the reconstructed digital video while at the same time preserving the high frequency details, thus avoiding degradation in video quality. The said apparatus is also simple in implementation and therefore, results only in small increase in cost when adding the said apparatus as part of a postprocessing module to the decompression system for the reconstruction of digital video data from compressed data.

The aim of this invention is to provide a very simple, and yet efficient method of removing the ringing effect around the edges in the decoded block. This method is suitable for hardware and application specific silicon chip implementation.

## SUMMARY OF THE INVENTION

The object of the present invention is therefore to provide a video signal decoding apparatus which solves these problems.

In order to achieve the aforementioned objective, a video signal decoding apparatus for decoding a coded video signal produced by encoding an input video signal, said apparatus comprises a variable length and run-length decoding means for decoding said coded bit to reconstruct said blocks of quantized coefficients; an inverse quantization means for inverse quantizing said blocks of reconstructed quantized coefficients; an inverse coding means for transforming said inverse quantized blocks of reconstructed quantized coefficients to produce a decoded block; an edge block detection means for determining whether said decoded block is classified as an edge block or not; and an adaptive edge enhancement filter means for removing the distortion from said decoded block when said decoded block is determined as an edge block by said block edge detec-

tion means.

### BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects and features of the present invention will become clear from the following description taken in conjunction with the preferred embodiments thereof with reference to the accompanying drawings throughout which like parts are designated by like reference numerals, and in which:

Fig 1 is a block diagram showing a video signal decoding apparatus according to a first embodiment of the present invention.

Fig. 2 is a block diagram showing an edge detector used in the video signal decoding apparatus of Fig. 1.

Figs. 3A and 3B are graphs showing the pixel labelling performed by an adaptive edge enhancement filter used in the video signal decoding apparatus of Fig. 1.

Fig. 4 is a flow chart showing the operation of a ringing noise filter used in the video signal decoding apparatus of Fig. 1.

Fig. 5 is block diagram showing a video signal encoder for compressing a digital video signal into an encoded video signal.

Fig. 6A is a table showing coefficients of a coded edge block according to the present invention.

Fig. 6B is a graph showing a waveform corresponding to the coded edge block of Fig. 6A according to the present invention.

Fig. 7A is a table showing coefficients of a non edge block.

Fig. 7B is a table showing coefficients of the non edge block that is produced by transforming the block of Fig. 7A.

Fig. 8A is a table showing coefficients of an edge block.

Fig. 8B is a table showing coefficient of the edge block that is produced by transforming the block of Fig. 8A.

Fig. 9A is a table showing coefficients of an ideal vertical edge block.

Fig. 9B is a graph showing a waveform of the vertical edge block of Fig. 9A.

Fig. 10A is a table showing coefficients of edge block that is produced by decoding the ideal vertical edge block of Fig. 9A.

Fig. 10B is a graph showing a wave form of the decoded edge block of Fig. 10A.

Fig. 11 is a block diagram showing a ringing noise removing apparatus according to a second embodiment of the present invention.

Fig. 12 is a block diagram showing a ringing noise detector used in the ringing noise removing apparatus of Fig. 11.

Fig. 13 is a block diagram showing a pixel level com-

parator used in the ringing noise source detector of Fig. 12.

Fig. 14 is a block diagram showing a ringing noise filter used in the ringing noise removing apparatus of Fig. 11.

Fig. 15 is a block diagram showing a construction of pixel selectors incorporated in the ringing noise filter of Fig. 14.

Fig. 16 is a table showing the decoding logic for a divisor generator incorporated in the ringing noise filter of Fig. 14, with respect to input data selected by the pixel selectors of Fig. 15.

Fig. 17 is a block diagram showing a divider incorporated in the ringing noise filter of Fig. 14.

Fig. 18 is graph showing a waveform of an original video data frequency response at an object's edges prior to compression and reconstruction, and

Fig. 19 is a graph showing a waveform of video data that is reconstructed from the video data of Fig. 18.

### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

(First Embodiment)

Before describing a video signal decoding apparatus according to the present invention, an encoding of a digital video signal indicating images, by compressing thereof, into a bit stream is described first. In Fig. 5, a video signal encoder Ev for compressing a digital video signal Sv into an encoded video signal Sc that is bit streams of compressed video signal Sv is shown. Note that the video signal decoding apparatus of the present invention is used for decoding that encoded video signal Sc, and applies the post-processing thereto to enhance the decoded images.

The video signal encoder Ev includes a frame memory 131, a block sampler 102, a motion vector estimator 103, a motion compensator 104, a discrete cosine transformer 105 which is indicated as "DCT", a quantizer 106, a rate controller 107, a coding unit 108, an inverse quantizer 109, an inverse discrete cosine transformer 110 which is indicated as "inverse DCT", a local decoded frame memory 111, and a reference frame memory 112, connected to each other as best shown in Fig. 5. The video signal encoder Ev utilizes a DCT and motion estimation and compensation technique.

An input digital interlaced video sequence Sv first enters the frame memory 131 in a frame by frame basis. In a compression system involving motion estimation and compensation technique, the input video frames are classified into predictive frame (P-frame) or intra frame (I-frame).

The I-frame is coded independently of the other frames, i.e. it is compressed without the use of any other frame. The I-frame is used periodically to allow for easy random access, fast forward/reverse and easy error recovery.

The P-frame is coded with motion estimation and compensation using previously coded frame to predict the contents of the current frame to be coded. Motion estimation and compensation is used to reduce the temporal redundancy in the digital video sequences and thereby increase the compression ratio.

When an input digital frame  $S_v$  is fed into the frame memory 131, it is also passed to the reference frame memory 112 for the motion estimation process thereby. It is then passed to a block sampler 102. The block sampler 102 partitions the image into spatially non-overlapping blocks of pixel data. To provide a reasonable level of adaptiveness, a block size of  $8 \times 8$  pixel may be used.

When the frame is classified as the I-frame, the sampled blocks are then transferred to DCT 105 when a well known transform coding method, Discrete Cosine Transform (DCT), is performed. The pixel data is transformed to obtain block of DCT coefficients. Thus obtained DCT coefficients are then transferred to the quantizer 106.

The quantizer 106 produces the quantized coefficients. The Quantizer 106 uses a quantizer matrix and quantizer step size given by the rate controller 107. The rate controller 107 ensures that a fairly constant amount of data is generated per second. The quantized coefficients are then transferred to the coding unit 108 which performs a routine of the run-length coding and variable length coding of the quantized coefficients. The result of the Run-length Coding & Variable Length Coding by the coding unit 108 is the output as bit stream that is the video signal  $S_c$  bit stream to be stored or transmitted to the receiver. The coding unit 108 also transfers this bit stream to the rate controller 107.

The quantizer 106 also transfers the quantized coefficients to the inverse quantizer 109 which performs the reverse of quantization to obtain the inverse-quantization coefficients. These inverse-quantization coefficients are then transferred to Inverse DCT 110 for the inverse DCT process to obtain the inverse DCT coefficients. The inverse DCT coefficients are also the decoded pixel block. The decoded pixel block are then transferred to the local decoded frame memory 111 to form the reconstructed (also referred to as "decoded") frame. The reconstructed frame will be used by the motion compensator 104 to process for the pixel block of the P-frame.

In the case of P-frame; two additional processes (compared to those done on I-frame) are performed. The block of pixel data (block to be coded) from the block sampler 102 is transferred not to the DCT 105 but to the motion vector estimator 103.

The estimator 103 estimates the motion vector by finding the best matched block from the pixel data of a previous frame. The horizontal and vertical offset of the best matched block from the previous frame, which together from the motion vector, are obtained. The block of pixel data and its corresponding motion vector are then transferred to the motion compensator 104.

The motion compensator 104 obtains blocks of data (referred to as "predicted block") from the previously reconstructed frame based on the decoded frame output from the local decoded frame memory 111, by using the offset given by the motion vector. Differences between values of the block to be coded and the predicted block are computed to obtain a differential block composed of the different values.

When the energy (for example, calculating the standard deviation of the block) of this difference block is smaller than the energy of the block to be coded, the block will then be coded with motion compensation. Otherwise, the block to be coded will be coded as it is.

An information bit will be used to indicate whether the motion compensation is done to the block, or not. When the block is to motion compensated, this difference block is then transferred to the DCT 105 for the Discrete Cosine Transformation (DCT). Otherwise, the block to be coded will be transferred to the DCT 105. The other process are the same as the case of I-frame. The motion vector is also transferred as a side information, to the DCT 105, to Quantization 6, and to the coding unit 108, and is then multiplexed with the variable length coded information of the difference block.

Referring to Fig. 1, a video signal decoding apparatus according to a preferred embodiment of the present invention is shown. The video signal decoding apparatus  $D_v$  includes a decoding unit 141, an inverse quantizer 143, an inverse discrete cosine transformer 145 indicated as "inverse DCT", a switch 153 indicated as "SI", a motion compensator 149, an edge detector 158, an adaptive edge enhancement filter 162, a decoded frame memory 151, and a local decoded frame memory 155 indicated as "DECODED FRAME MEMORY".

The decoding unit 141 is connected to the coding unit 108 (Fig. 5) of the video signal encoder  $E_v$  for receiving the encoded video bit stream  $S_c$  therefrom. This bit stream  $S_c$  is the compressed video signal and also has a coded information. The decoding unit 141 applies the encoded video signal  $S_c$  with a Variable Length & Run-length Decoding operation. Then, decoding unit 141 demultiplexes and decodes the variable length codes to produce the quantized coefficients of each block of the coded frame and the side information of the block, i.e. the motion vector  $MV$  and the information bit on whether the block is motion compensated or not. The decoding unit 141 also checks the frame number of the frame to be decoded to see whether the frame is I-frame or P-frame.

The inverse quantizer 143 is connected to the decoding unit 141 for receiving the decoded quantized coefficient block therefrom to apply an inverse quantization process thereto.

The inverse DCT (inverse Discrete Cosine Transformer) 145 is connected to the inverse quantizer 143 for receiving the inverse quantized coefficient block therefrom to apply an inverse discrete cosine transformation process thereto. Thus, the inverse DCT 145 pro-

duces the inverse DCTed coefficient block that is a sequence of blocks of decoded pixel data.

The switch 153 is connected to the inverse DCT 145 for receiving the inverse DCTed coefficient block therefrom, and receives the side information of the block from the decoding unit 141 through the inverse quantizer 143 and the inverse DCT 145. The switch 153 checks the frame number thereof to determine whether the frame to be decoded is I-frame or P-frame. The switch 153 has a first output port connected to the motion compensator 149 by a line L148 and a second output port connected to the edge detector 158 by a line L147.

For the I- frames, the switch 153 sends the output from the inverse DCT 145 directly to the edge detector 158 through the line 147 for an edge detection process which is described in detail with reference to Fig. 4 later.

For the P-frames, the switch 153 checks the information bit indicating whether a block is coded with motion compensation or not and. When it is indicated that the block output from the inverse DCT 145 had been coded without motion compensation by an encoder such as typically shown in Fig. 5, the switch 153 sends that block directly to the edge detector 158 through the line L147, for which an edge detection process is again done on the blocks of decoded pixel.

When it is indicated that the block from the Inverse DCT circuit 145 had been coded with the motion compensation by such an encoder (Ev), the switch 153 sends that block to the motion compensator 149 through the line L148. It is because that the block of decoded pixel is the differential block and a predicted block has to be obtained with a motion compensation process performed in the motion compensator 149. In such case, the switch 153 sends the output of the inverse DCT 145 to the motion compensator 149.

On the other hand, the local decoded frame memory 155 is connected to the decoding unit 141 for receiving the decoded motion vector MV to produce a predicted block using the offset given by that decoded motion vector MV.

The motion compensator 149 is also connected to the decoded frame memory 155 for receiving the predicted block therefrom. The motion compensator 149 adds the difference block from the switch 153 to the predicted block from the memory 155, and produces a constructed block.

The edge detector 158 is also connected to the motion compensator 149 for receiving the reconstructed block therefrom to apply the edge detection process thereto.

The edge detector 158 checks whether the reconstructed block is a block containing edge information (edge block) or not (non-edge block). The edge detector 158 has a first output port connected to the adaptive edge enhancement filter 162 by a line L161 and a second output port connected to the decoded frame memory 151 by a line L160.

When it is an edge block, the edge detector 158

transfers the block to the adaptive edge enhancement filter 162 through the line L161. The adaptive edge enhancement filter 162 performs an adaptive filtering process to remove the ringing effect and generate the filtered block. Details of the adaptive edge enhancement filter 162 will be described in later with reference to Figs. 3A and 3B. However, when it is a non-edge block, the detector 158 transfers the block to the decoded frame memory 151 through the line L160.

10 The decoded frame memory 151 is also connected to the adaptive edge enhancement filter 162 by a line L163 for receiving the filtered block therefrom. The decoded frame memory 151 is further connected to the decoded frame memory 155 for outputting the data stored therein as an output reconstructed sequence therefrom. The output reconstructed sequence will be used for motion compensation of the next frame.

15 The edge detector 158 includes a mean calculator 170, a maximum and minimum calculator 171 each connected to the motion compensator 149 and the switch 153 for receiving the decoded block therefrom, as described in the above. The mean calculator 170 calculates the mean value  $V_{MEAN}$  of the block, and the maximum and minimum calculator 171 calculates the maximum value  $V_{MAX}$  and the minimum value  $V_{MIN}$  of the decoded block.

20 The edge detector 158 further includes a decider 172 connected to both the mean calculator 170 and the maximum and minimum calculator 171 for receiving the calculated values  $V_{MEAN}$ ,  $V_{MAX}$ , and  $V_{MIN}$  therefrom. The decider 172 obtains a first absolute difference  $Da1$  between the maximum value  $V_{MAX}$  and the mean value  $V_{MEAN}$  and a second absolute difference  $Da2$  between the minimum value  $V_{MIN}$  and the mean value  $V_{MEAN}$ .

25 Then, the decider 172 compares thus obtained absolute differences  $Da1$  and  $Da2$  with a first threshold  $Th1$  that is a predetermined value suitable for determining the block to be smoothed, and is chosen to "48" in this embodiment. When either of the first and second absolute differences  $Da1$  and  $Da2$  is greater than the first threshold  $Th1$ , the decider 172 judges that the decoded block is an edge block. Otherwise, the decider 172 judges the decoded block as a non-edge block. According to this judgment, the decider 172 selectively transfers the decoded blocks either of the adaptive edge enhancement filter 162 and the decoded frame memory 151, as described in the above.

30 With reference to Figs. 3A and 3B, the pixel labelling performed by the adaptive edge enhancement filter 162 of Fig. 1 are described briefly. In Fig. 3A, the blocks of pixel on the left is the 8 x 8 decoded block which has been classified as an edge block is shown. In Fig. 3B, the post-processed block of the edge block of Fig. 3A is shown. The adaptive filter is applied to the pixel values of the decoded block one at a time, starting from the top-left corner, going horizontally and ending at the bottom-right corner.

35 For every pixel value, the corresponding filtered val-

ue is stored in the processed edge block of Fig. 3B. For example, a pixel at the 3rd row and the 4th column of the edge block of Fig. 3A, which is represented by P0, is targeted, the adaptive filter uses its surrounding pixel (P1, P2, P3, P8) to generate a post-processed value S0 and stored in the processed edge block (Fig. 3B). Note that each of 9 pixels having a target pixel P0 and surrounding 8 pixels can be represented by reference Pi, wherein "i" is a positive integer smaller than 9 and including 0, as shown.

Referring to Fig. 4, an operation of the adaptive edge enhancement filter 162 is shown. When every edge block enters into the adaptive edge enhancement filter 162, the adaptive edge enhancement routine starts, for processing the edge block on a pixel-by-pixel basis. Each pixel is adaptive filtered using the surrounding pixels.

At step S1, it is judged whether any pixel not processed is rested in the edge block or not. When all pixels in the block have been processed, it is judged as "NO". Then the procedure terminates. However, since the newly entered block has pixels not processed, it is judged "YES". Then, the procedure advances to step S3 for being processed.

At step S3, all of parameters "i", "sum", and "count" are set to 0. Thus, the center pixel P0 shown in Fig. 3A is set as the target pixel for the processing. The procedure advances to next step S5.

At step S5, it is judged whether "i" is smaller than 9 or not. When "i" is smaller than 9, it means that any of surrounding 8 pixels is still not examined. Then, it is judged as "NO" and the procedure advances to step S7.

At step S7, it is judged whether the difference between the center pixel P0 and its surrounding pixel Pi is smaller than a second threshold Th2 having a predetermined value. When the difference (Pi - P0) is smaller than the second threshold Th2, meaning that the center pixel P0 and that particular surrounding pixel Pi are on the same side of the edge in the image. Then, it is judged as "YES", the procedure advances to step S11.

At step S11, the value of count is increased by 1 and the value of Pi is also added to sum. Then the procedure advances to step S9.

At step S9, "i" is incremented by 1, and then the procedure returns to step S5. Thus, the relationship between the center pixel P0 and the next surrounding pixel Pi is examined.

At step S7, when it is judged as "NO", it means that the difference (Pi - P0) is equal to or greater than the second threshold Th2. The procedure also advances to step S9.

However, when "i" becomes 9 as a result of addition at step S9, it is judged as "YES" at step S5. Then the procedure advances to step S13. Thus, the loop of step S5, S7, S11, and S9 is repeated until when all the surrounding eight pixels are scanned.

At step S13, it is judged whether the value of count is equal to either of 0, 1, 3, and 7, or not. When it is

judged "YES", the procedure advances to step S19.

At step S19, the count is incremented by 1, and the center pixel P0 is added to the sum. Then, the procedure advances to step S25.

5 However, when it is judged as "NO" at step S13, it is meant that count is equal to neither of 0, 1, 3, and 7. The procedure advances to step S15.

At step S15, it is judged whether value of count is equal to 5 or not. When the count is equal to 5, it is 10 judged as "YES". Then the procedure advances to step S21.

At step S21, the count is incremented by 3. The three times of the center pixel P0 is added to the sum. Then the procedure advances to step S25.

15 However, when it is judged as "NO" at step S15, the procedure advances to step S17.

At step S17, it is judged whether the value of count is equal to 6 or not. When it is judged as "YES", the procedure advances to step S23.

20 At step S23, the count is increased by 2 and two times of the center pixel P0 is added to the sum. Then, the procedure advances to step S25.

However, at step S17, it is judged as "NO", the procedure advances to step S25.

25 At step S25, the post-processed block S0 is calculated by taking the division of sum by count. In this embodiment, the second threshold Th2 is set to be at 16, but can be chosen to any predetermined value enabling surrounding pixels on the same side of the edge to be smoothed. Then, the procedure advances to step S27.

At step S27, the post-processed block S0 is put to process edged block, and then the procedure returns to step S1 to examine the next block.

As apparent from the above description, by performing the adaptive smoothing function on the detected edge blocks, the ringing (corona) effects has been reduced significantly. The subjective quality of the decoded video sequences, after passing through the post-processing filter, has shown improvement compared to those without using the post-processing filter.

Referring to Figs. 6A and 6B, the post-processed edge block according to the present invention and its corresponding waveform are shown, respectively. As best shown in Fig. 6B, the post-processed block has a significantly reduced ringing effect. With reference to the MPEG, JPEG and H.261 coding standard, performing the smoothing function on the coded picture means that no extra information needs to be transmitted by the coder to the decoder and hence no change in the bit-syntax is needed.

55 Thus, the effects of applying this method is that corona noise around the edge is reduced significantly, no further information to be transmitted to the decoder, and also no change to the existing MPEG, JPEG and H.261 coding standards is required. By using a simple edge detection method that is tightly coupled to the simple adaptive averaging operation to process the edge block, hardware implementation of the method is also very

simple.

According to the present invention, the first step in the operation is to take the decoded block from the decoder and pass it through an edge deduction circuit to determine if the decoded block contains any edge information. If edge information is present in the block, the block is classified as edge block; otherwise, it is classified as non-edge block.

Then, the edge block is passed through a smoothing filter circuit which performs an averaging process adaptively, using only neighboring pixel from the same side of the edge for the averaging process. The neighboring pixel are classified as to be on the same side of the edge as the pixel to be smoothed or on different side of the edge using a thresholding technique. If the value of neighboring pixel differs from the pixel to be smoothed by less than the threshold value, it is classified as to be on the same side of the edge as the pixel to be smoothed; otherwise, it is on the different side of the edge to be smoothed. The averaging process then uses only those neighboring pixel which are on the same side as the pixel to be smoothed for the calculation of the average value.

#### (Second Embodiment)

Referring to Fig. 11, a ringing noise removing apparatus according to a second embodiment of the present invention is shown. In Fig. 11, the ringing noise removing apparatus Rn includes a video data decoder 1, a ringing noise source detector 2, a status memory 3, a reorder memory 4, a ringing noise filter 5, and a post processing unit 6.

The video data decoder 1 is provided for receiving a compressed video data Sc from a suitable data source such as a digital video cassette recorder. The video data decoder 1 decompresses the compressed data Sc to reconstruct the original video data Sp. This reconstructed data Sp is output from the video data decoder 1 at random and at high transfer rate, and is arranged in a block format having a matrix formed by 8 x 8 pixels, for example.

The ringing noise source detector 2 is connected to the video data decoder 1 for receiving the video pixel data Sp therefrom to detect the ringing noise components included in the data Sp to produce a first detection level signal S<sub>DS1</sub> and a writing command signal S<sub>W</sub>.

The status memory 3 is connected to the ringing noise source detector 2 for receiving the signals S<sub>DS1</sub> and S<sub>W</sub> therefrom to store the detection status of the pixel data.

The reorder memory 4 is connected to the video data decoder 1 for receiving the pixel data Sp therefrom. The reorder memory 4 reorders frames of the pixel data Sp to convert thereof from the block format into a raster scan format. Thus, the pixel data Sp is converted into a raster scan pixel data Sr suitable for raster scanning and other decoder system's purpose.

The ringing noise filter 5 is connected to the status memory 3 and the reorder memory 4 for receiving the signals (data) S<sub>DS2</sub> and Sr therefrom, respectively. The ringing noise filter 5 filters a ringing noise components 5 from the data Sr with respect to the signal S<sub>DS2</sub> and further produces the reading command signal S<sub>RD</sub>. The ringing noise filter 5 is further connected to the status memory 3 for feeding the signal S<sub>RD</sub> back thereto. Then, the ringing noise filter 5 outputs a ringing noise filtered data S<sub>F</sub> whose ringing noise components are removed therefrom.

The post processor 6 is connected to the ring noise filter 5 for receiving the filtered pixel data S<sub>F</sub> to apply post-processing such as interpolation, and produces a pixel data Sp' which is the reproduction of the original pixel data.

Referring to Fig. 12, the ringing noise source detector 2 incorporated in the ringing noise removing apparatus Rn is shown. The detector 2 includes a First In First Out buffer memory 7, hereinafter referred to as "FIFO buffer", connected to the video data decoder 1 for receiving the reconstructed pixel data Sp therefrom. The FIFO buffer 7 outputs thus received pixel data Sp therefrom in the same order in which the FIFO buffer 7 received. Furthermore, the FIFO buffer 7 produces a flag signal Fs indicating whether the pixel data Sp is current present therein, or not.

A pixel level detection controller 8 is connected to the FIFO buffer 7 for receiving the flag signal Fs to check the status of FIFO buffer 7. The controller 8 produces a reading signal Sr1 and the writing command signal S<sub>W</sub> when the flag signal Fs indicates that the FIFO buffer 7 current has data Sp therein. The pixel level detection controller 8 is also connected to the FIFO buffer 7 for transferring the reading command signal Sr1 thereto so as to outputs the reconstructed data Sp therefrom.

A pixel level detector 50 is connected to the FIFO buffer 17 and the pixel level detection controller 8 for receiving the reconstructed data Sp and the reading command signal Sr1 therefrom, respectively. The pixel level detector 50 detects the pixel level of the reconstructed data Sp outputted from the FIFO buffer with respect to the reading command signal Sr1. The pixel level detector 50 includes a maximum pixel comparator 9, a mean block calculator 10, and minimum pixel comparator 11 each connected to the FIFO buffer 7 and the pixel level detection controller 8, respectively, as best shown in Fig. 12.

The maximum pixel comparator 9 checks and compares all pixels in the reconstructed block data Sp to detect the maximum value thereof, and further produces a first level signal S<sub>MAX</sub> with respect to the reading signal Sr1. The reading signal Sr1 is used as a clock signal for reading operation. Similarly, the minimum pixel comparator 11 checks and compares all pixels in the reconstructed block data Sp for the minimum value, and produces a second signal S<sub>MIN</sub>. The mean block calculator 10 calculates an average value of the pixels in the block

$S_p$ , and produces a third signal  $S_{MEAN}$ . Note that the reading signal  $S_{r1}$  is used as an output clock for the FIFO buffer 7 and pixel level detector 50.

Specifically, when the detection controller 8 had read one block of pixel data  $S_p$  from the FIFO buffer 7 by the reading signal  $S_{r1}$ , the reading cycle will stop and each data at the output of comparators 9, 10, and 11 is the final output for that block. Thus, three level signals  $S_{MAX}$ ,  $S_{MIN}$ , and  $S_{MEAN}$  are produced, and are transferred from the level detector 50.

A pixel level comparator 12 is connected to the pixel level detector 50 for receiving the level signals  $S_{MAX}$ ,  $S_{MIN}$ , and  $S_{MEAN}$  therefrom. Based on these level signals, the pixel level comparator 12 completes the final computation to determine the detection status of the block data  $S_p$  being examined, and produces a first detection level signal  $S_{DS1}$ .

Referring to Fig. 13, the pixel level detector 12 is shown. The pixel level detector 12 includes a first subtracter 13, a second subtracter 14, a first comparator 15, a pixel level detection threshold setter 16, and a second comparator 17. The threshold setter 16 produces a threshold signal  $S_{TH}$  having a predetermined value, or can store the threshold signal  $S_{TH}$  supplied from an external means such as a keyboard operated by an operator.

The first subtracter 13 is connected to the maximum pixel comparator 9 and the mean block calculator 10 of the pixel level comparator 12 for receiving the signals  $S_{MAX}$  and  $S_{MEAN}$  therefrom, respectively. The second subtracter 14 is connected to the mean block calculator 10 and the minimum pixel comparator 11 of the pixel level comparator 12 for receiving the signals  $S_{MEAN}$  and  $S_{MIN}$  therefrom, respectively. Note that the mean value is essentially less than or equal to the maximum value, and is greater than or equal to the minimum value of the data block pixels  $S_p$ . Therefore, the first subtracter 13, subtracts the mean value signal  $S_{MEAN}$  from the maximum value signal  $S_{MAX}$  to produce a first difference signal  $S_{d1}$ . Similarly, the second subtracter 14 subtracts the minimum value signal  $S_{MIN}$  from the mean value signal  $S_{MEAN}$  to produce a second difference signal  $S_{d2}$ .

The first comparator 15 is connected to the subtracters 13 and 14 for receiving the difference signals  $S_{d1}$  and  $S_{d2}$ , respectively, to compare thereof. The first comparator 15 multiplexes either one of the two difference signals  $S_{d1}$  and  $S_{d2}$  having a greater value to output as a comparison signal  $S_{DC}$  therefrom. The comparison signal  $S_{DC}$  is a direct current.

The second comparator 17 is connected to the first comparator 15 and the pixel level detection threshold setter 16 for receiving the signal  $S_{DC}$  and the threshold signal  $S_{TH}$  therefrom, respectively. The comparator 17 compares the comparison signal  $S_{DC}$  with the threshold signal  $S_{TH}$  to produce and output the first detection level signal  $S_{DS1}$  therefrom.

According to this embodiment, when this comparison result signal  $S_{DC}$  is greater than the threshold signal

$S_{TH}$ , it is judged that the block of pixels being examined have ringing noise to be filtered around the high frequency areas thereof. In this sense, the first detection level signal  $S_{DS1}$  can be regarded as a pixel status signal indicating the status of the ringing noise in the block pixels.

Referring back to Fig. 11, The status memory 3 is connected to both the pixel level comparator 12 and the pixel level detection controller 8 of the ringing noise detector 2 for receiving the first detection level signal  $S_{DS1}$  and the writing command signal  $S_w$  therefrom, respectively. The status memory 3 stores the signal  $S_{DS1}$  together with the writing command signal  $S_w$  therein. The operation of the status memory 3 will be described later with respect to the ringing noise filter 5 below.

Referring to Fig. 14, details of the ringing noise filter 5 of Fig. 11 are shown. The circuitry is synchronous and is clocked by the operation clock that is a vertical synchronization signal  $V. sync$ . The ringing noise filter 5 includes first and second line memories 18 and 19 each having a memory length equal to the number of pixel data in one raster scan line.

The first line memory 18 is connected to the reorder memory 4 for receiving the raster scan pixel data  $S_r$  to delay by one line, and then produces a first delayed pixel data  $S_{r1}$ . This data  $S_{r1}$  is delayed from the pixel data  $S_r$  by one line period.

The second line memory 19 is connected to the first line memory 18 for receiving the first delayed pixel data  $S_{r1}$  therefrom to also delay by one line, and then produces a second delayed pixel data  $S_{r1}'$ . This data  $S_{r1}'$  is delayed from the pixel data  $S_r$  by one line period.

Four 8-bit registers, entitled as DEF 21, 24, 25, and 26 in Fig. 14, are provided. The first resistor 21 is connected to the reorder memory 4 for receiving the raster scan pixel data  $S_r$  to delay by one operation clock, and then produces a third delayed data  $S_{rc}$ . This data  $S_{rc}$  is delayed from the pixel data  $S_r$  by one clock period.

The second resistor 24 is connected to the first line memory 18 for receiving the second delayed pixel data  $S_{r1}$  therefrom to delay by one clock, and then produces a fourth delayed pixel data  $S_{r1c}$ . This data  $S_{r1c}$  is delayed from the data  $S_r$  by one line and one clock period.

The third resistor 25 is connected to the second resistor 24 for receiving the fourth delayed pixel data  $S_{r1c}$  therefrom to delay thereof by one clock, and then produces a fifth delayed pixel data  $S_{r1c}'$ . This data  $S_{r1c}'$  is delayed from the data  $S_r$  by one line and two clocks period.

The fourth resistor 26 is connected to the second line memory 19 for receiving the second delayed pixel data  $S_{r1}'$  to delay thereof by one clock, and then produces a sixth delayed pixel data  $S_{r1c}'$ . This data  $S_{r1c}'$  is delayed from the data  $S_r$  by two lines and one clock period.

It is to be noted that the fifth delayed pixel data  $S_{r1c}'$  indicates the pixel data to be filtered and the center pixel of a filter matrix, and is referred to as "a center pixel signal CPX". Specifically, pixels contributing to the computation of the filtering of the center pixel are one line

ahead of it at the output of the fourth resistor 26, one clock ahead at the output of the third resistor 25, one clock delayed at the line memory 18 output and one line delayed at the output of DFF 21.

Four pixel multiplexers that are entitled as PMUX 22, 23, 27 and 28, respectively, in Fig. 14 are provided. These multipliers 22, 23, 27 and 28 have an identical circuitry, as typically shown in Fig. 15. The first pixel multiplexer 22 is connected to the first and second registers 21 and 24 for receiving the third delayed pixel data  $S_{rc}$  and the center pixel signal  $CPX$  ( $S_{rlc}$ ), and produces a first comparison signal  $Sc1$ .

Referring to Fig. 15, the construction of first pixel multiplier 22 is shown, as a representative example of the multipliers 22, 23, 27 and 28. The pixel multiplier 22 includes a filter threshold setter 33, an absolute difference calculator 34, a multiplexer 35, and a comparator 36. The filter threshold setter 33 produces a threshold signal  $S_{TF}$  having a predetermined value, or can store the threshold signal  $S_{TF}$  supplied from an external means such as a keyboard operated by an operator.

The absolute difference calculator 34 is connected to the first resistor 21 and the second resistor 24 for receiving the third delayed pixel data  $S_{rc}$  and the center pixel signal  $CPX$  therefrom, respectively. The calculator 34 calculates an absolute difference between the two data  $S_{rc}$  and  $CPX$  to produce a third difference signal  $Sd3$ .

The comparator 36 is connected to the filter threshold setter 33 and the absolute difference calculator 34 for receiving the filter threshold signal  $S_{TF}$  and the third difference signal  $Sd3$  therefrom, respectively. The comparator 36 compares these signals  $S_{TF}$  and  $Sd3$  to produce the first comparison signal  $Sc1$  having two levels "0" and "1". For example, when the absolute difference  $Sd3$  is greater than the threshold  $S_{TF}$ , the comparison signal  $Sc1$  has "1" level, and otherwise "0" level.

The multiplexer 35 is connected to the first register 21 and the comparator 36 for receiving the third delayed pixel data  $S_{rc}$  and the first comparison signal  $Sc1$  therefrom, respectively. When the absolute difference  $Sd3$  is greater than the filter threshold  $S_{TF}$ , the multiplexer 35 outputs the pixel data  $S_{rc}$ . However, when the absolute difference  $Sd3$  is not greater than the threshold  $S_{TF}$ , the multiplexer 35 outputs a zero value signal.

Referring back to Fig. 14, a summation unit 29 is connected to the multiplexer 35 of the first pixel multiplexer 22 for receiving the signal  $S_{rc}$  or the zero value signal therefrom. A divisor generator 31 is connected to the comparator 36 of the first pixel multiplexer 22 for receiving the first comparison signal  $Sc1$  therefrom.

Since each of the pixel multiplexers 23, 27, and 28 has a construction identical to that of the pixel multiplexer 22, as described in the above, the operation thereof is substantially the same. Therefore, only the portions of multiplexers 23, 27, and 28 different from the first multiplexer 21 are described below.

The second pixel multiplexer 23 is connected to the

first line memory 18, the second register 24, the summation unit 29, and the divisor generator 31, as shown in Fig. 14. Similarly, the second pixel multiplexer 23 produces a second comparison signal  $Sc2$ , and outputs the first delayed pixel data  $S_{rl1}$  or the zero value signal according to the second comparison signal  $Sc2$ .

The third pixel multiplexer 27 is connected to the second line memory 19, the second register 24, the summation unit 29, and the divisor generator 31. Similarly, the third pixel multiplexer 27 produces a third comparison signal  $Sc3$ , and outputs the sixth delayed pixel data  $S_{rl1}'$  or the zero value signal according to the third comparison signal  $Sc3$ .

The fourth pixel multiplexer 28 is connected to the third register 25, the second register 24, the summation unit 29, and the divisor generator 31. Similarly, the fourth pixel multiplexer 28 produces a fourth comparison signal  $Sc4$ , and outputs the fifth delayed pixel data  $S_{rl1}'$  or the zero value signal according to the third comparison signal  $Sc4$ .

Thus, the divisor generator 31 receives the four comparison signals  $Sc1$ ,  $Sc2$ ,  $Sc3$ , and  $Sc4$  from the four pixel multipliers 22, 23, 27, and 28, respectively. Based on these four signals, the divisor generator 31 produces a divisor signal  $Sdv$ .

Referring to Fig. 16, a logic table of the divisor generator 31 is shown. Since each of comparison signals  $Sc1$ ,  $Sc2$ ,  $Sc3$ , and  $Sc4$  can be "0" or "1" as described in the above, a summed value  $S_{cs}$  of the four comparison signals  $Sc1$ ,  $Sc2$ ,  $Sc3$ , and  $Sc4$  becomes either of numbers "0", "1", "2", "3", and "4". When the summed value  $S_{cs}$  is "0", the divisor generator 31 outputs "1" as the divisor  $Sdv$ . Similarly, when the summed value  $S_{cs}$  is "1", "2", and "3 or 4", the divisor  $Sdv$  is set to "1", "2", and "4", respectively.

Also, the summation unit 29 receives the pixel data  $S_{rc}$  (or zero),  $S_{rl1}$  (or zero),  $S_{rl1}'$  (or zero), and  $S_{rl1}'$  (or zero) from the multipliers 22, 23, 27, and 28, respectively, and sums thereof up to produce a summation signal  $S_s$ .

A divider 32 is connected to the summation unit 29, the divisor generator 31, and the second register 24 for receiving the summation signal  $S_s$ , the divisor signal  $Sdv$ , and the center pixel signal  $CPX$ , respectively, therefrom. Based on these three signals, the divider 32 produces a division signal  $Sdiv$ .

Referring to Fig. 17, details of the divider 32 is shown. The divider 32 includes an adder 37 and a dividing computer 38 connected to each other as shown. The adder 37 is connected to the second register 24 and the summation unit 29 for receiving the center pixel signal  $CPX$  and the summation signal  $S_s$ , respectively, therefrom to add thereof.

The dividing computer 38 has four input ports IN1, IN2, IN3, and IN4. The input ports IN1 and IN2 are connected to the adder 37 for receiving the added single  $CPX$  and  $S_s$  therefrom. The input ports IN3 and IN4 are connected to the summation unit 29 for receiving the

summation signal  $S_s$  therefrom. The dividing computer 38 selects one of signals that are taken therein through the input ports IN1, IN2, IN3, and IN4 based on the divisor signal  $S_{dv}$ , and further outputs thus selected signal therefrom as the division signal  $S_{dv}$ .

Within the block of the dividing computer 38, the dividing logic thereof is schematically shown. When the divisor signal  $S_{dv}$  is "0", the signal input through the input port IN1 is selected and output as the division signal  $S_{dv}$ . Similarly, the divisor signals  $S_{dv}$  is "1", "2", or "4", the input signals through the ports IN2, IN3, and IN4 are output, respectively, from the divider 32. Thus, depending on the divisor signal  $S_{dv}$  from the divisor generator 31, the selected bits from either the adder 37 or the summation 29 will be output from the divider 32.

Referring back to Fig. 14, a circular buffer controller 20 is connected to the status memory 3, and operates bases on a vertical synchronization signal  $V_{Sync}$  supplied thereto. The circular buffer controller 20 generates a read signal  $S_{RD}$  including the memory address and reading command so as to read the second detection level signal  $S_{DS2}$  from the status memory 3 as well as to keep track of the position of the pixel being filtered.

A data multiplexer 30 is connected to the status memory 3 (Fig. 11), the second register 24, and the divider 32 for receiving the second detection level signal  $S_{DS2}$ , the central pixel signal  $CPX$ , and the division signal  $S_{dv}$ , respectively, therefrom. The data multiplexer 30 multiplexes the division signal  $S_{dv}$  with the central pixel data  $CPX$  by the second detection level signal  $S_{DS2}$ .

Thus, the data multiplexer 30 will output the data  $S_{dv}$  from the divider 32 when the detection level (or status) signal  $S_{DS2}$  indicates presence of ringing noise within that pixel ( $CPX$ ) boundaries or else the pixel value remains unchanged from previous i.e. prior to input of the ringing noise filter 5. The data that is outputted by the multiplexer 30 will then be the final output of the ringing noise filter 5, and is the filtered pixel data  $S_F$ .

As is apparent from the above descriptions, the ringing noise removing apparatus of the present invention is capable of maintaining the reconstructed video sharpness while removing the ringing noise effect from the reconstructed vide by detecting the sources causing such noise and applying an adaptive filter to the detected areas. Also, according to the present invention, complex mathematical calculations are not necessary, and only an independent process is required. Thus, makes it fairly easy to implement in conjunction with a video decoder (or decompressor). Hence, the apparatus of the present invention has a large effect in removal of the ringing noise form reconstructed video at low cost and complexity.

By the above described configuration, the ringing noise removing apparatus reads in the reconstructed digital video data in a block by block basis from the decompress (or decoder). Each block is then detected for the high frequency areas which is a potential source of

the ringing noise. The frequency level detected depends on the detection threshold which can be adjusted. The block detection status generated is then stored. This detection status is used for filtering control.

5 In the meantime, the reconstructed video data is converted from block format into raster scan through the reorder memory. After the pixel data is converted to raster scan, it is output to the ringing noise filter. The pixel to be filtered is assumed to be in the center of a 3 x 3 matrix. All the pixels within the matrix is then compared with an adjustable threshold. Pixels within a certain range are then selected for the computation of the filter result. Using the pixels selected, the filter result of the particular pixel is then calculated. The filtered pixel is then multiplexed with the original pixel using the detection status which was stored earlier. The detection status act as the control for the multiplex to produce the final output from the apparatus.

20 Although the present invention has been fully described in connection with the preferred embodiments thereof with reference to the accompanying drawings, it is to be noted that various changes and modifications are apparent to those skilled in the art. Such changes and modifications are to be understood as included within the scope of the present invention as defined by the appended claims unless they depart therefrom.

## Claims

30 1. A video signal decoding apparatus for decoding a coded video signal produced by encoding an input video signal, said apparatus comprising:

35 a variable length and run-length decoding means for decoding said coded bit to reconstruct said blocks of quantized coefficients; an inverse quantization means for inverse quantizing said blocks of reconstructed quantized coefficients;

40 an inverse coding means for transforming said inverse quantized blocks of reconstructed quantized coefficients to produce a decoded block;

45 an edge block detection means for determining whether said decoded block is classified as an edge block or not; and an adaptive edge enhancement filter means for removing the distortion from said decoded block when said decoded block is determined as an edge block by said block edge detection means.

50 2. A video signal decoding apparatus for decoding a coded video signal produced by encoding an input video signal with an input video signal with an encoding apparatus comprising:

a partitioning means for partitioning each frame of said input video signal into smaller blocks of pixel data;

a transform coding means for transforming said partitioned blocks into blocks of transformed coefficients;

a quantization means for quantizing said transformed coefficients to produce transformed coefficients: and

a run-length and variable coding means for run-length and variable length coding said quantized coefficients to produce a coded bit.

said video signal decoding apparatus comprising:

a variable length and run-length decoding means for decoding said coded bit to reconstruct said blocks of quantized coefficients;

an inverse quantization means for inverse quantizing said blocks of reconstructed quantized coefficients;

an inverse coding means for transforming said inverse quantized blocks of reconstructed quantized coefficients to produce a decoded block;

an edge block detection means for determining whether said decoded block is classified as an edge block or not: and

an adaptive edge enhancement filter means for removing the distortion from said decoded block when said decoded block is determined as an edge block by said block edge detection means.

3. A video signal decoding apparatus as claimed in Claim 2, wherein said edge block detection means comprises:

a first calculating means for calculating an average value of said decoded block;

a second calculating means for calculating a maximum value and a minimum value of said decoded block;

a third calculating means for calculating a first absolute difference value between said average value and said maximum value;

a fourth calculating means for calculating a second absolute difference value between said average value and said minimum value: and

a determining means for determining that said decoded block is an edge block when either of said first and second absolute difference values is greater than a predetermined value, and otherwise determine that said decoded block is a non-edge block.

4. A video signal decoding apparatus as claimed in Claim 2, wherein said adaptive edge enhancement filter means performs convolution on a pixel by pixel

5 bases on the decoded edge block with an adaptive averaging filter mask by comparing a difference between every neighboring pixel of said pixel and said pixel to be convolved with a second predetermined value: and averaging by said neighboring pixel when said difference is judged as smaller than said second predetermined value.

10 5. A video signal decoding apparatus as claimed in Claim 2, wherein said encoding apparatus further compares a first motion compensation means for removing the temporal redundancy and likewise in said partitioned blocks before being transformed by said transform coding means; and said video decoding apparatus further comprising a second motion compensation means for performing an inverse motion compensation of said reconstructed block before being processed by said edge block detection means.

15 6. A video signal decoding apparatus for decoding a coded video signal, said removing ringing noise in a decoded image pixel data comprising:

20 a decoding means for decoding said coded video signal to reconstructed image pixel data and partitioning said reconstructed image pixel data into blocks of pixels;

a ringing noise source detection means for comparing said reconstructed image pixel data with a predetermined value to detect the source and location of the ringing noise in a block by block basis in said reconstructed image pixel data and to produce a status signal;

a memory means for said status signal therein: and

25 a ringing noise filter means for filtering the ringing noise from said reconstructed image pixel data based on said status signal.

30 35 40 45 50 55

Fig. 1

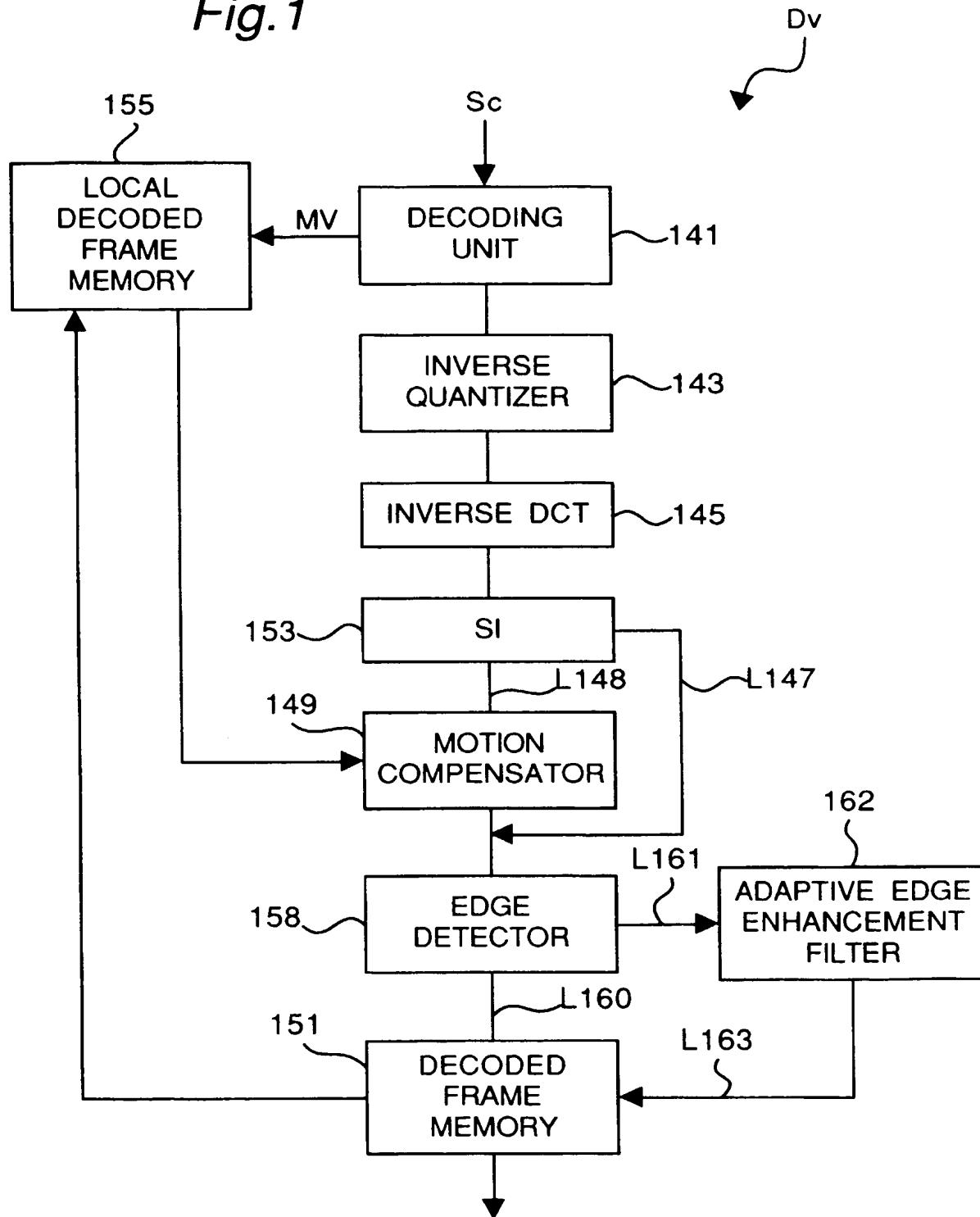


Fig.2

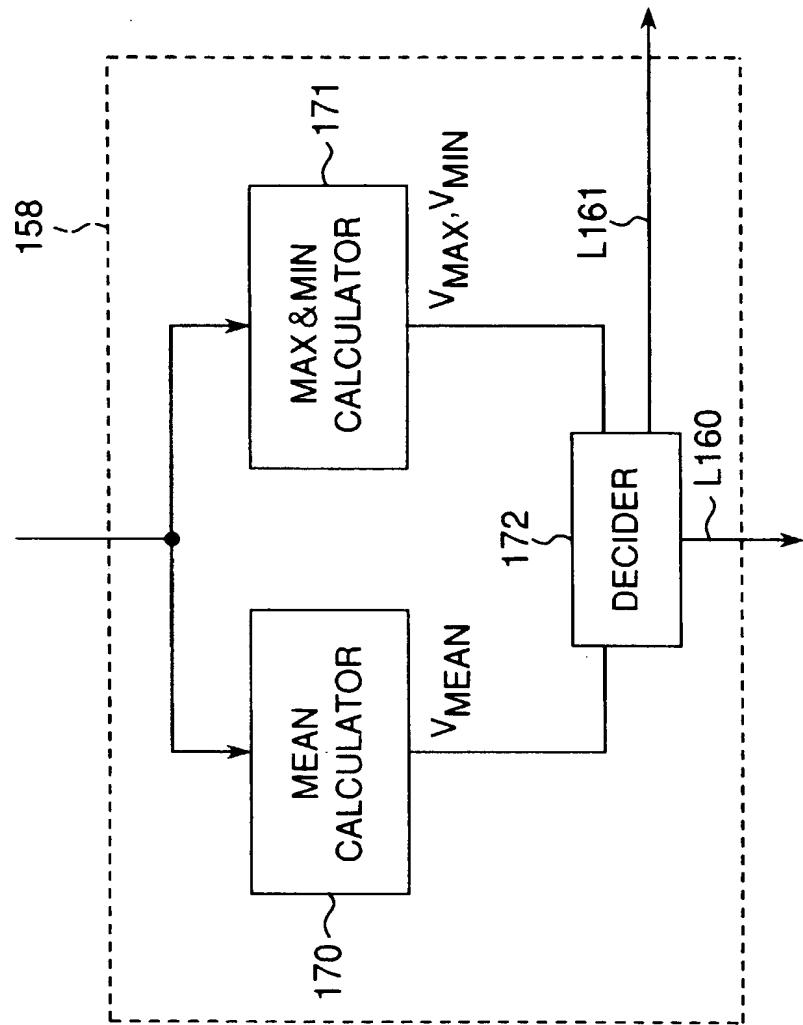


Fig.3A

P1	P2	P3
P4	P0	P5
P6	P7	P8

Fig. 3B

so

Fig.4

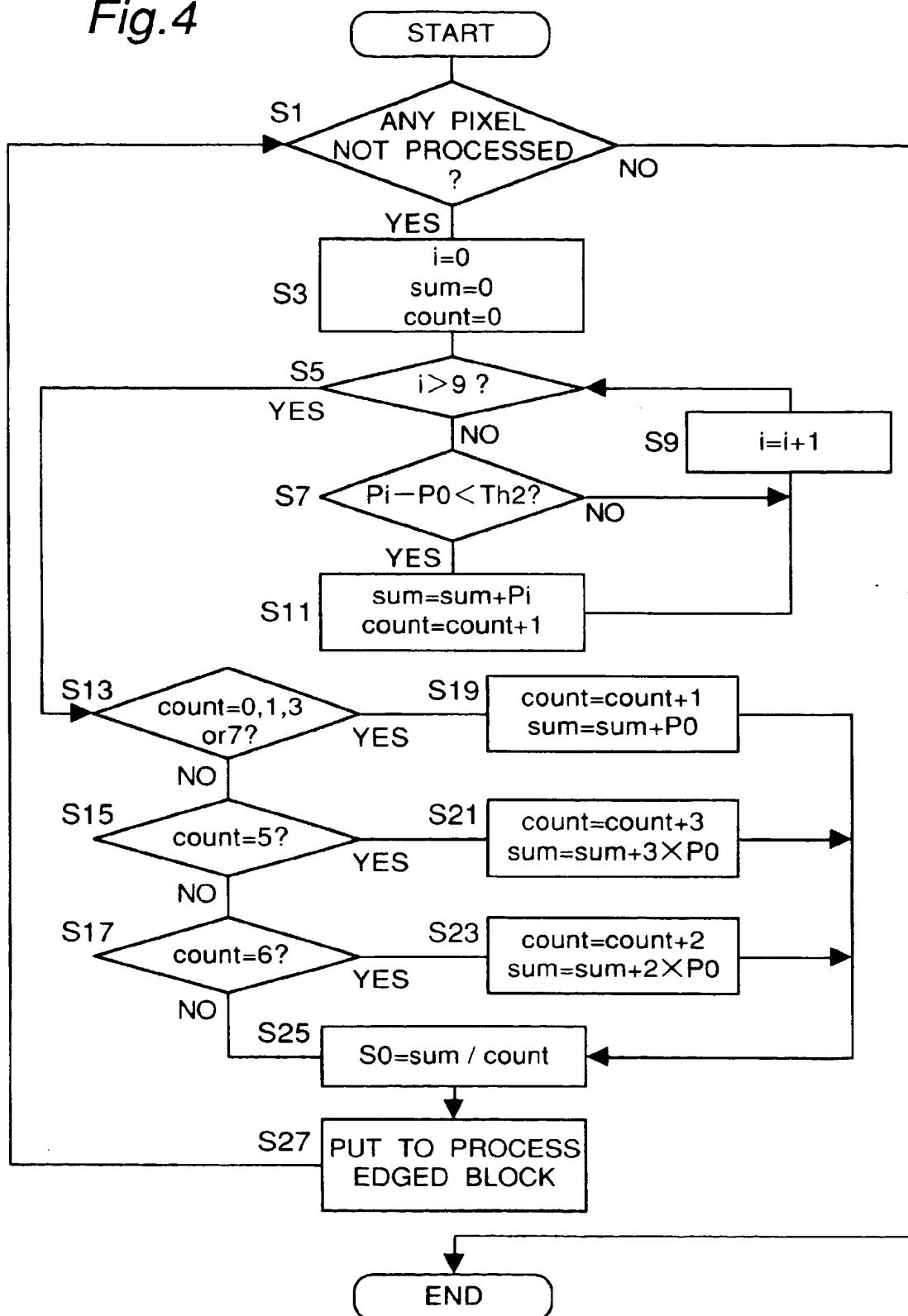


Fig.5

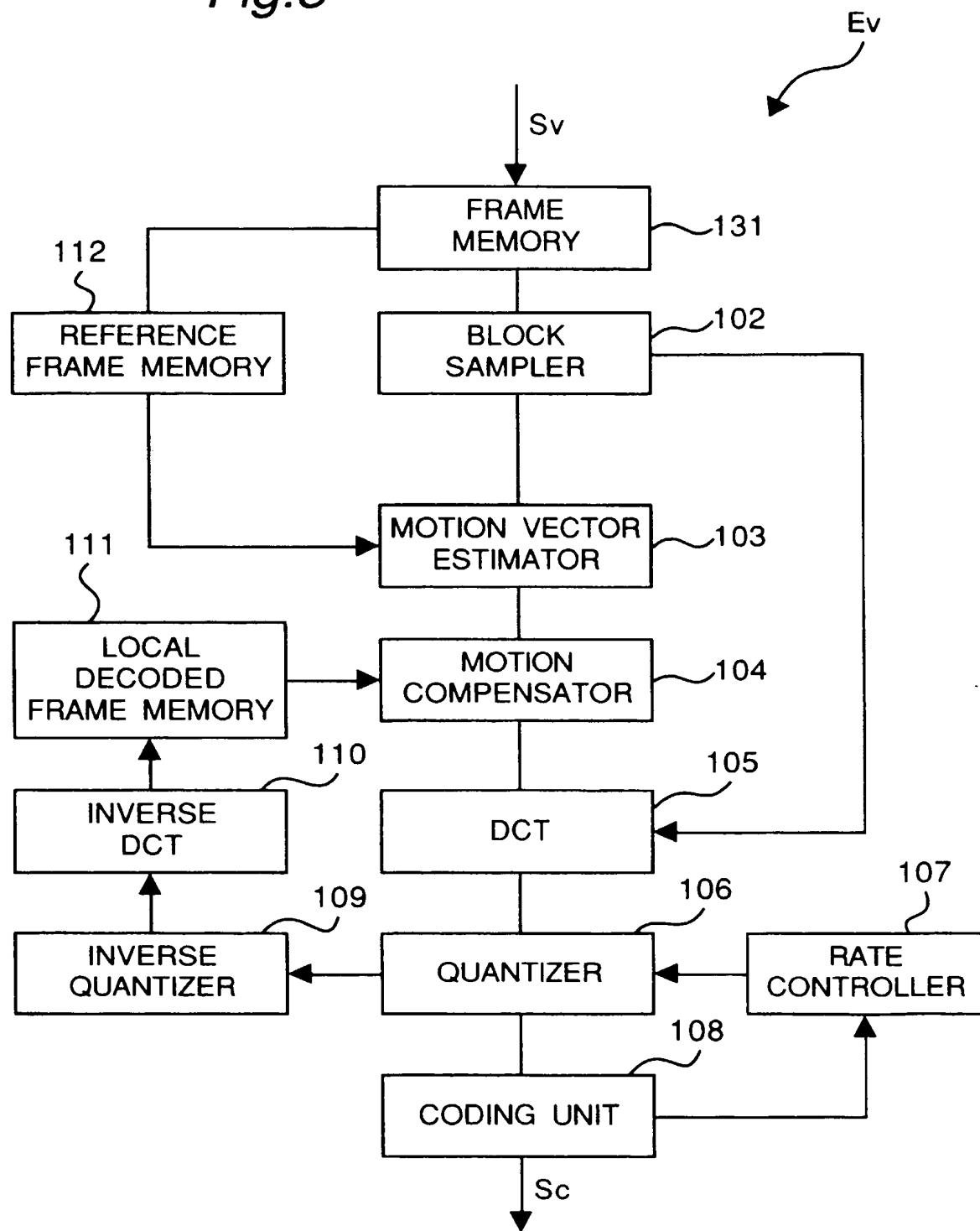


Fig. 6A

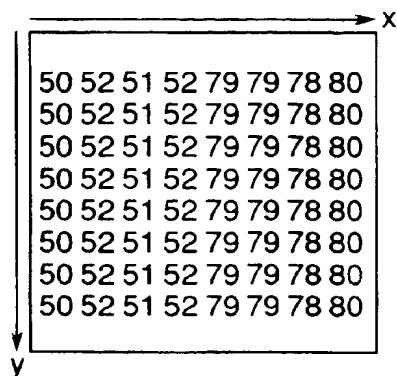


Fig. 6B

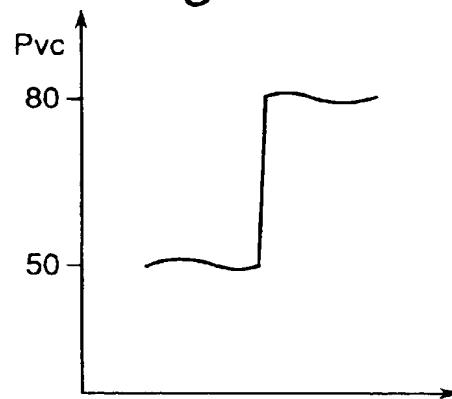


Fig. 9A

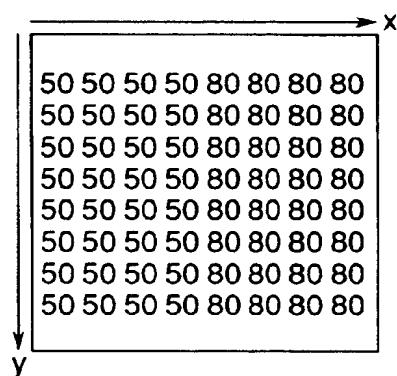


Fig. 9B

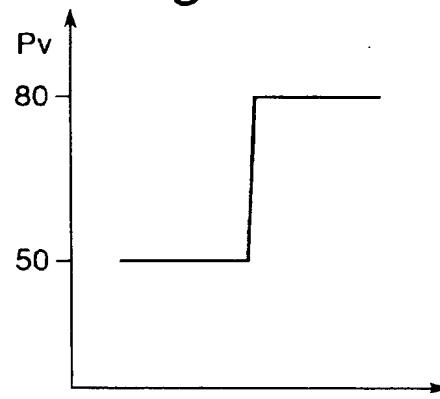


Fig. 10A

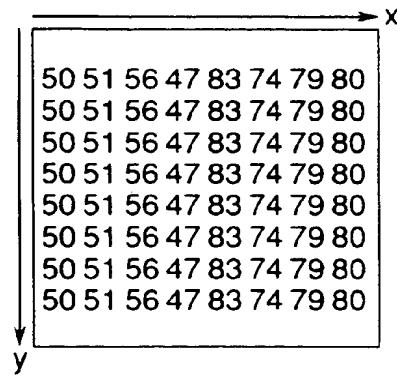


Fig. 10B

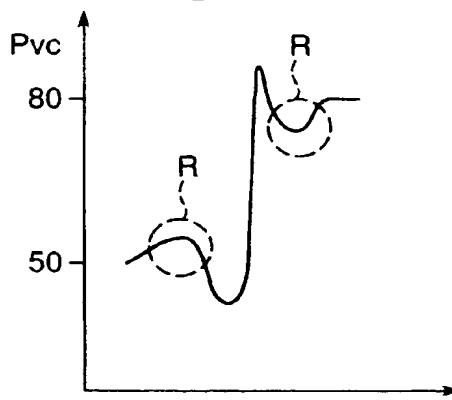


Fig. 7A

104	106	109	107	104	101	99	98
105	107	104	102	97	100	99	94
102	102	99	97	98	96	94	95
98	98	97	98	92	89	93	91
96	93	96	93	88	89	88	85
90	91	90	84	88	85	84	81
86	87	88	84	81	85	83	79
83	79	79	82	83	82	82	82

Fig. 7B

740	20	-1	-1	-4	2	1	1
60	6	-2	-3	-1	0	-2	-2
0	-5	-2	1	-1	-1	2	0
3	4	-4	-1	-2	0	1	1
1	-3	-2	-3	3	3	1	1
1	0	-2	-3	1	0	-1	0
-1	-2	0	1	3	-4	0	3
1	0	-1	0	0	0	3	-1

Fig. 8A

244	232	241	233	222	207	211	207
233	190	176	210	217	228	210	211
74	70	76	118	220	234	216	210
170	194	181	100	52	188	237	210
243	230	241	88	70	227	223	208
236	232	222	249	90	73	236	217
242	222	254	156	41	200	232	204
247	229	243	212	44	129	245	206

Fig. 8B

1527	0	189	-56	-106	104	23	-28
20	-108	-134	117	76	-100	-1	28
111	78	-53	-24	58	32	-19	-7
109	133	24	-53	7	18	-7	5
27	63	63	-22	-70	4	26	0
-38	-30	4	-11	-36	-1	10	-4
-41	-46	-58	-33	72	6	-67	17
-19	-60	-22	69	-35	-46	61	17

Fig. 11

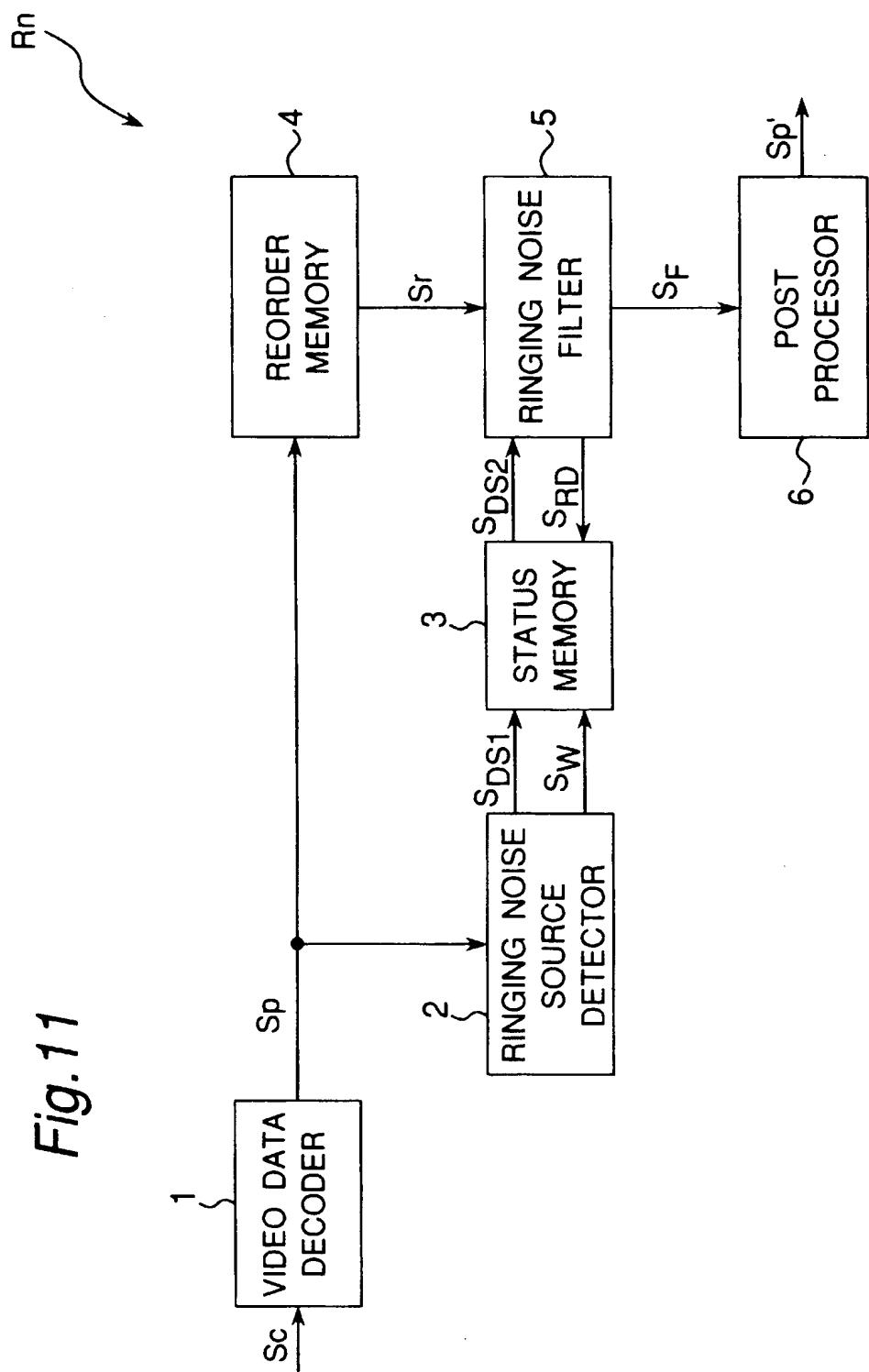


Fig. 12

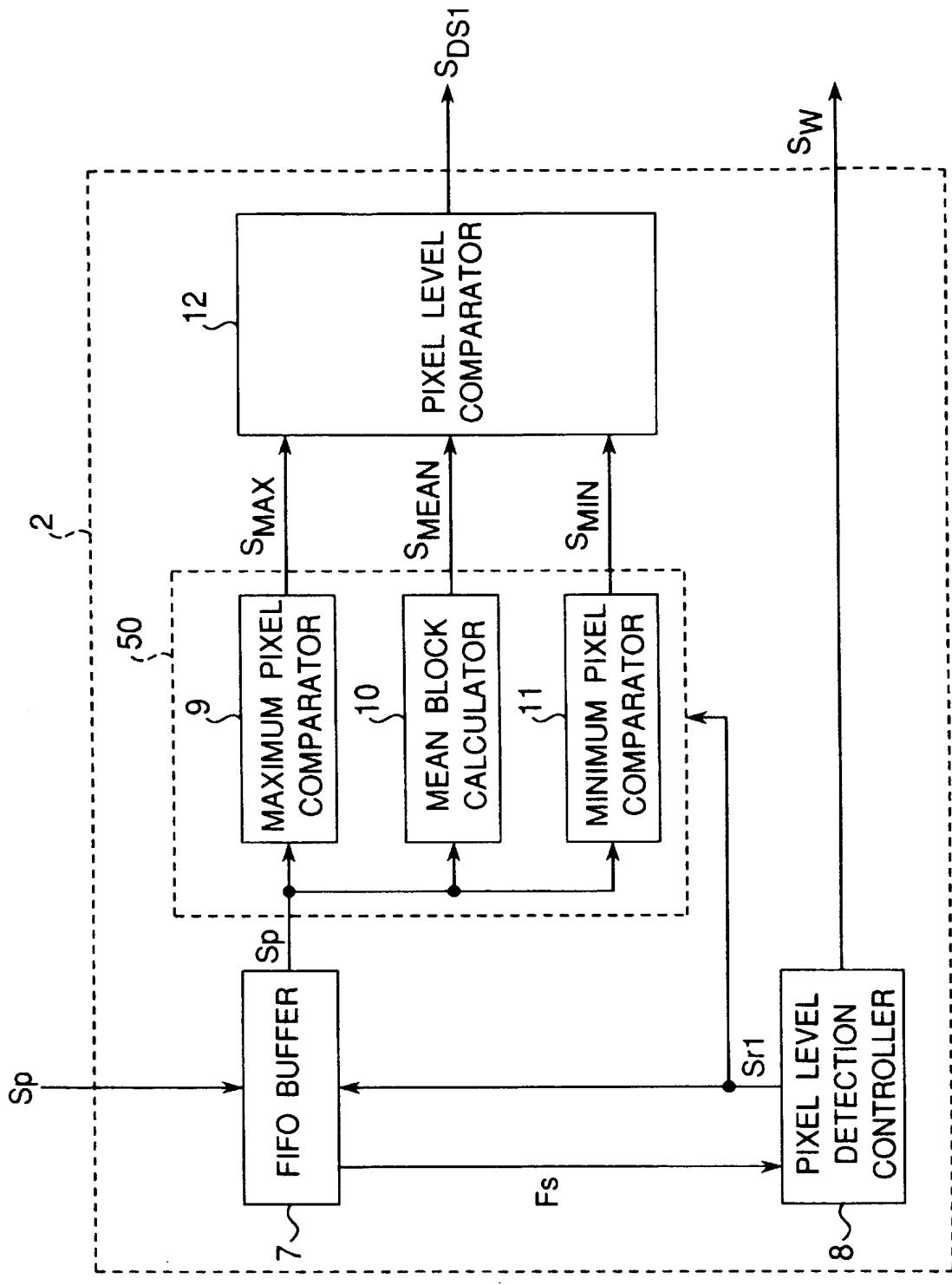


Fig. 13

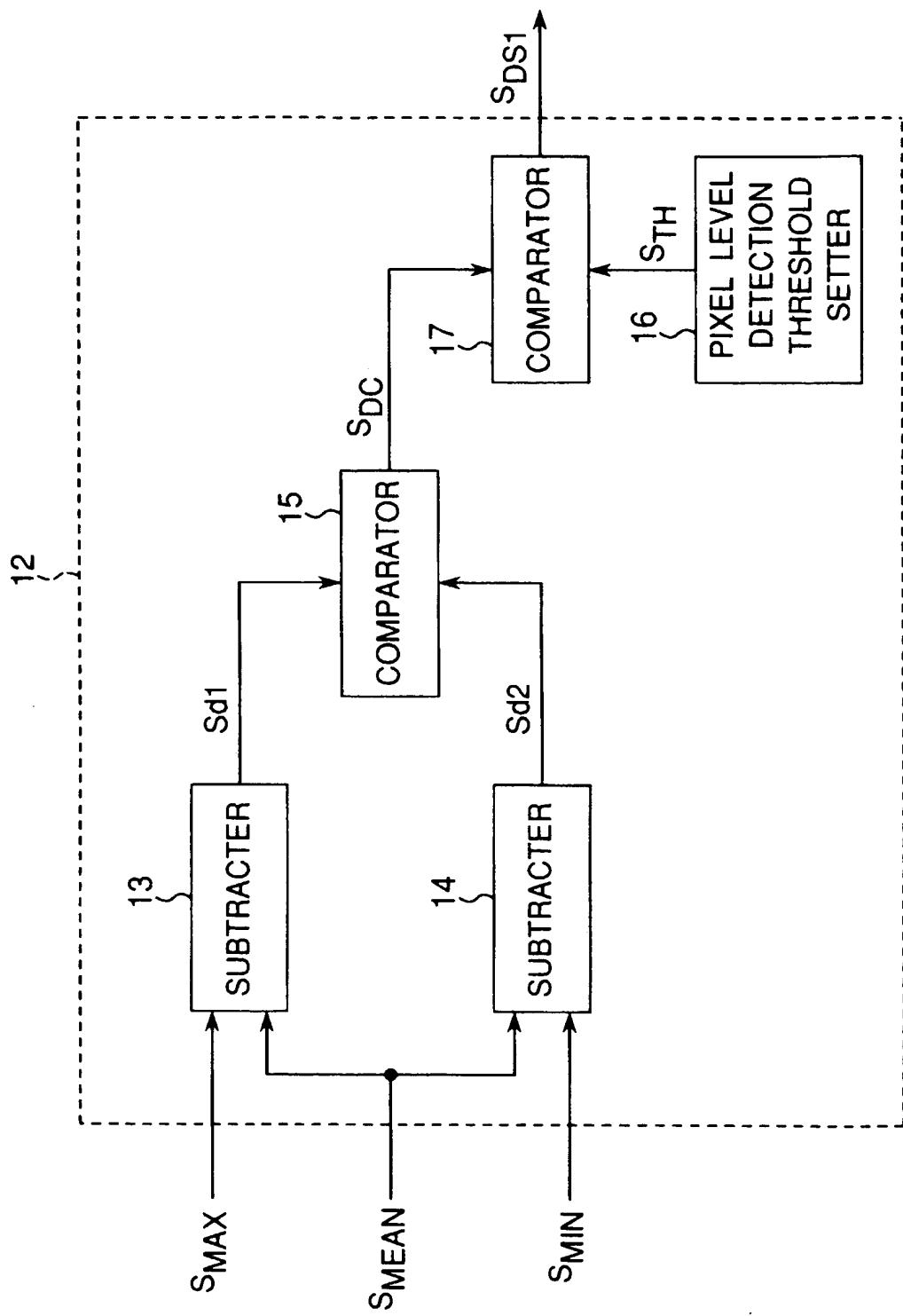


Fig. 14

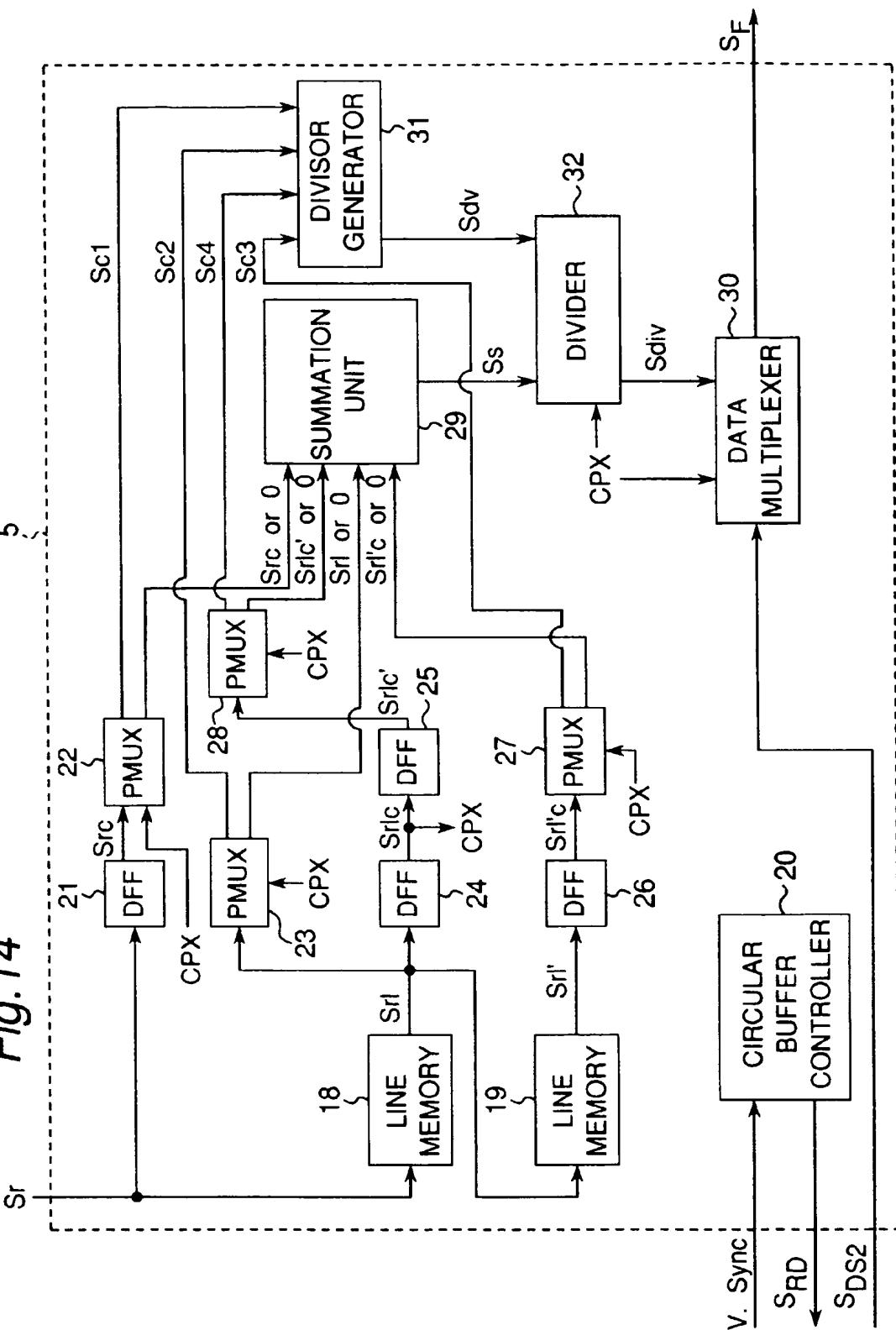


Fig. 15

22,23,27,28

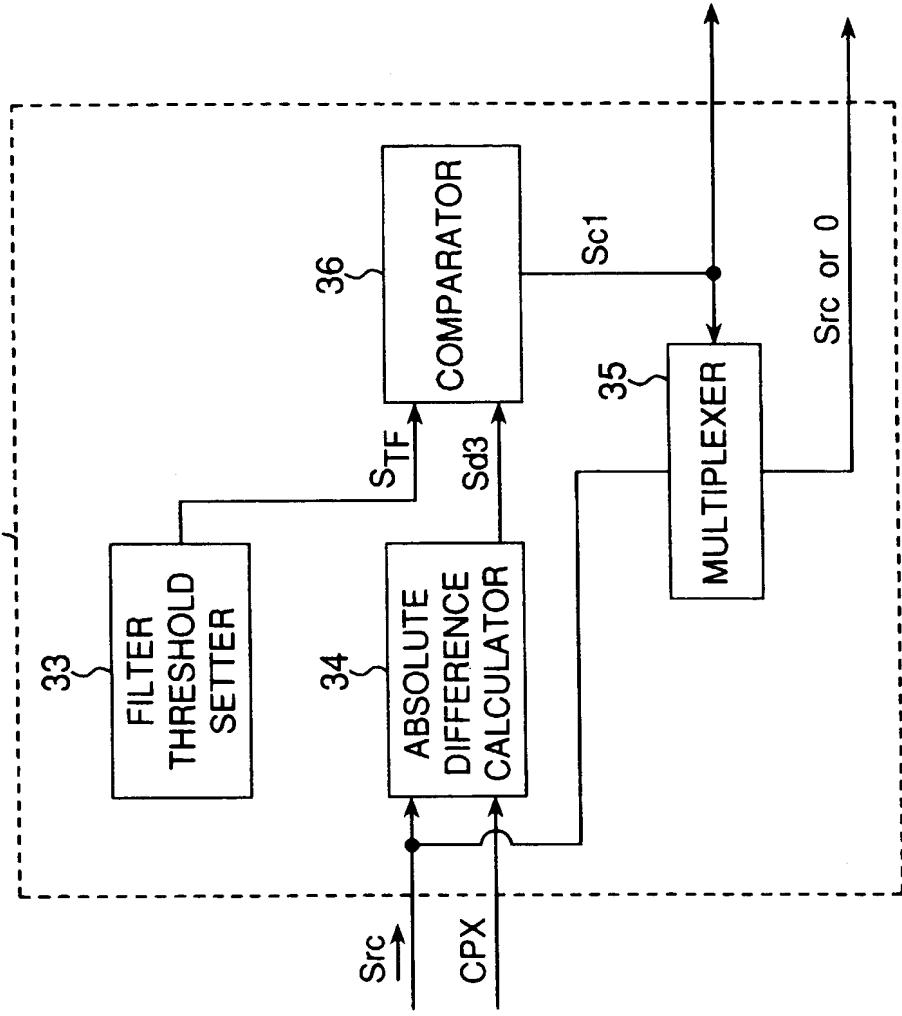


Fig. 16

Scs	Sdv
0	1
1	1
2	2
3	4
4	4

Fig. 17

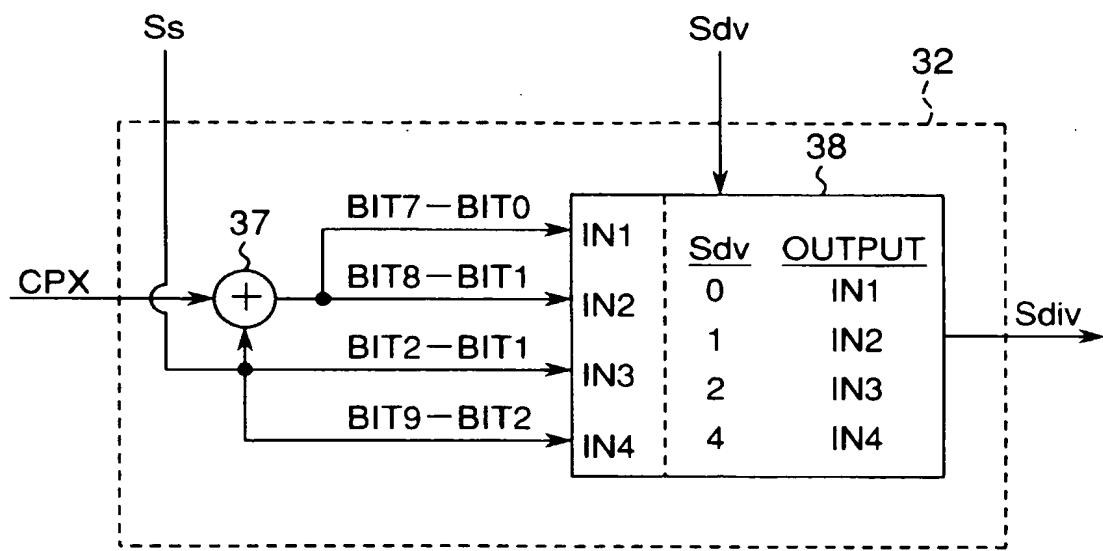


Fig. 18

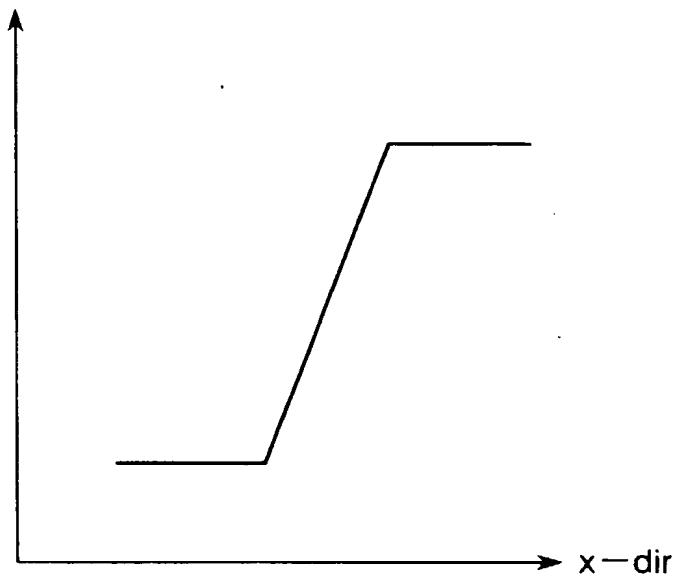


Fig. 19

